



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/679,008	10/02/2003	Robert C. Chang	SANDP036	8904
67674 7590 01/16/2008 ANDERSON, LEVINE & LINTEL fbo SANDISK CORPORATION 14785 PRESTON ROAD SUITE 650 DALLAS, TX 75254			EXAMINER TSAI, SHENG JEN	
			ART UNIT 2186	PAPER NUMBER
			MAIL DATE 01/16/2008	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/679,008

Applicant(s)

CHANG ET AL.

Examiner

Sheng-Jen Tsai

Art Unit

2186

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 09 November 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,3-5,10,15,19,22 and 43-50 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1,3-5,10,15,19,22 and 43-50 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 02 October 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- ☐ Notice of Informal Patent Application
- ☐ Other: _____

DETAILED ACTION

1. This Office Action is taken in response to Applicant's Amendments and Remarks filed on November 9, 2007 regarding application 10/679,008 filed on October 2, 2003.

2. Claims 1, 3, 22 and 48 have been amended.

Claims 49-50 have been added.

Claims 2, 6-9, 11-14, 16-18, 20-21 and 23-42 have been cancelled.

Claims 1, 3-5, 10, 15, 19, 22 and 43-50 are pending for consideration.

3. ***Response to Amendments and Remarks***

Applicants' remarks have been fully and carefully considered, with the Examiner's response set forth below.

(1) Applicants amend independent claims 1 and 22 to recite the new sequence of write operations to be "writing contents of a first logical sub-unit into the first programming unit in a first group of programming units;" "then writing contents of a third logical sub-unit into the third programming unit in a first group of programming units;" and "then writing contents of a second logical sub-unit into a programming unit in a second group of programming unit."

Thus, as currently amended, claims 1 and 22 require a sequence of write operations in the order of (1) writing contents of a first logical sub-unit, (2) then writing contents of a third logical sub-unit, and (3) then writing contents of a second logical sub-unit.

Applicants point to paragraphs [0046] and [0050] of Applicants' Patent Application Publication (US Patent Application Publication 2004/0103241) for support of

the new sequence of write operations, and contend that the Miyauchi reference (US 6,430,650) fails to teach this newly recited sequence of write operations.

The Examiner has carefully and thoroughly reviewed paragraphs [0046] and [0050] of Applicants' Patent Application Publication, and was not able to identify or locate any passages in these paragraphs that recite such a sequence of write operations.

It is noted that paragraph [0046] recites "By way of example, when contents 316 associated with a third page of the logical block (not shown) which corresponds to block 300 are to be written into block 300, contents 316 are written into a third page 304g which is included in the next available group 308, i.e., group 308b."

Paragraph [0047] then recites "When contents 312 associated with the first page of the logical block (not shown) are to be superceded by updated contents 312' ... ;" and

Paragraph [0048] then recites "For example, when contents 318, which are associated with the second page of the logical block (not shown) are to be written, contents 318 are stored into a second page 204n of fourth group 308d."

Note that paragraphs [0046] through [0048] describe the write operations in a general manner without any particular sequence by using the word "when," as opposed to the word "then." Even, for the sake of discussion and without conceding the point, assuming that the scenario of "when" as described in paragraphs [0046] to [0048] represents a sequence, the sequence would be (1) writing contents of a third logical sub-unit (as in paragraph [0046]), (2) then writing contents of a first logical sub-unit (as in paragraph [0047]), and (3) then writing contents of a second logical sub-unit (as in

paragraph [0048]). It is clear that the sequence of write operations recited in paragraphs [0046] to [0048] has a different order as recited in claims 1 and 22, hence a different sequence of write operations.

A close examination of paragraph [0050] reveals that it recites merely "contents 316 associated with a third page of the logical block are written after contents 314 associated with a fourth page of the logical block are written." Thus, paragraph [0050] does not provide support for the sequence of write operations as recited in claims 1 and 22, either.

Therefore, the newly amended sequence of write operations recited in claims 1 and 22 lacks support from the written description. As such, it should be removed from the claims.

(2) It should be pointed out that, with respect to the newly amended sequence of write operations, the Miyauchi reference does teach "out-of-sequence" write operations into two different groups of programming units.

First, Applicants already admitted that "the Miyauchi reference expressively teaches the performing of out-of-sequence writes within a single block" [page 11, lines 6-7, the REMARK section of Applicants' reply filed on 11/09/2007].

Second, the "out-of-sequence writes" operations in Miyauchi's invention are not limited to a single block. Figure 5 of Miyauch explicitly show a plurality of groups (i.e., blocks) of programming units. For example, group PGN=X comprises four programming unites denoted as 0, 1, 2, and 3. Similarly, group PGN=Y comprises four programming unites denoted as 0, 1, 2, and 3. The write operations is further illustrated in figures 7A

and 7B to explain when a logical unit of a “out-of-sequence writes” operations is destined in either group PGN=X or group PGN=Y [for example, figure 7A, step S104, “search a group not used;” figure 7B, step S120, “search a group not used”].

(3) Applicants also amend claims 1 and 22 with the additional limitation of “MLC NAND flash non-volatile memory,” admit that the Kramer reference indeed teaches the element of “MLC NAND flash non-volatile memory,” and contend that the combination of the Miyauchi and Kramer references is not proper because there is no motivation for combining them. The Examiner disagrees.

Kramer teaches that the motivation of using a MLC NAND flash non-volatile memory is to take advantage the benefits that it offers [A flash memory with multilevel cell significantly reduces the memory per-bit cost. For instance, a 32 Mb multilevel cell (MLC) Flash memory storing two bits of data per cell achieves 32 Mb memory storage capacity using 16M Flash memory cells (column 2, lines 5-10); a 128 Mb multi-level NAND flash memory stores two bits per cell by tight programmed cell threshold voltage (V_{th}) control. Program states have 0.4V V_{th} distribution and a 0.8V separation gap. Incremental-step pulse programming (ISPP) gives narrower V_{th} distribution with a smaller stepping voltage (column 2, lines 39-44)].

Therefore, it would have been obvious to use a MLC NAND flash non-volatile memory as the type of flash memory of Miyauchi's invention because it reduces the memory per-bit cost and gives narrower V_{th} distribution with a smaller stepping voltage.

(4) Therefore, the Examiner's position regarding the patentability of all claims remains the same as stated in the previous office Action.

Another iteration of claim analysis based on previously relied on references (Miyauchi, US 6,430,650 and Kramer, US 6,182,239) has been made. Refer to the corresponding sections of the following claim analysis for details.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. Claims 1, 3-5, 10, 15, 19, 22 and 43-50 rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Applicants amend independent claims 1 and 22 to recite the new sequence of write operations to be “writing contents of a first logical sub-unit into the first programming unit in a first group of programming units;” “then writing contents of a third logical sub-unit into the third programming unit in a first group of programming units;” and “then writing contents of a second logical sub-unit into a programming unit in a second group of programming unit.”

Thus, as currently amended, claims 1 and 22 require a sequence of write operations in the order of (1) writing contents of a first logical sub-unit, (2) then writing contents of a third logical sub-unit, and (3) then writing contents of a second logical sub-unit.

Applicants point to paragraphs [0046] and [0050] of Applicants' Patent Application Publication (US Patent Application Publication 2004/0103241) for support of the new sequence of write operations, and contend that the Miyauchi reference (US 6,430,650) fails to teach this newly recited sequence of write operations.

The Examiner has carefully and thoroughly reviewed paragraphs [0046] and [0050] of Applicants' Patent Application Publication, and was not able to identify or locate any passages in these paragraphs that recite such a sequence of write operations.

It is noted that paragraph [0046] recites "By way of example, when contents 316 associated with a third page of the logical block (not shown) which corresponds to block 300 are to be written into block 300, contents 316 are written into a third page 304g which is included in the next available group 308, i.e., group 308b."

Paragraph [0047] then recites "When contents 312 associated with the first page of the logical block (not shown) are to be superceded by updated contents 312' ... ;" and

Paragraph [0048] then recites "For example, when contents 318, which are associated with the second page of the logical block (not shown) are to be written, contents 318 are stored into a second page 204n of fourth group 308d."

Note that paragraphs [0046] through [0048] describe the write operations in a general manner without any particular sequence by using the word "when," as opposed to the word "then." Even, for the sake of discussion and without conceding the point, assuming that the scenario of "when" as described in paragraphs [0046] to [0048] represents a sequence, the sequence would be (1) writing contents of a third logical

sub-unit (as in paragraph [0046]), (2) then writing contents of a first logical sub-unit (as in paragraph [0047]), and (3) then writing contents of a second logical sub-unit (as in paragraph [0048]). It is clear that the sequence of write operations recited in paragraphs [0046] to [0048] has a different order as recited in claims 1 and 22, hence a different sequence of write operations.

A close examination of paragraph [0050] reveals that it recites merely "contents 316 associated with a third page of the logical block are written after contents 314 associated with a fourth page of the logical block are written." Thus, paragraph [0050] does not provide support for the sequence of write operations as recited in claims 1 and 22, either.

Therefore, the newly amended sequence of write operations recited in claims 1 and 22 lacks support from the written description. As such, it should be removed from the claims.

Claims 3, 5, 10, 43-45 and 49 are rejected by virtue of their dependency from claim 1.

Claims 19, 46-48 and 50 are rejected by virtue of their dependency from claim 22.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 1, 3-5, 10, 15, 19, 22 and 43-50 are rejected under 35 U.S.C. 103(a) as being unpatentable over Miyauchi (US 6,430,650), and in view of Kramer (US 6,182,239).

As to claim 1, Miyauchi discloses **a method** [Semiconductor Storage Device for Group management of Data Sectors (title); abstract] **for performing a write operation** [figures 7A and 7B show the flowchart of write operation] **in a non-volatile memory system** [The corresponding non-volatile memory system is the flash memory system shown in figure 1, 11, which includes a flash memory (figure 1, 15) which is non-volatile], **the non-volatile memory system including an MLC NAND flash non-volatile memory** [The corresponding non-volatile memory system is the flash memory system shown in figure 1, 11, which includes a flash memory (figure 1, 15) which is non-volatile; the MLC NAND aspect is taught by Kramer, see below] **having a plurality of erase units** [FIG. 2 shows an erase block 20 in a flash memory 15. A block size is 64 kilobytes. ... (column 3, lines 66-67)); the total capacity of the storage device is more than 1 megabytes (column 1, lines 54-60; column 3, lines 42-50), thus it has a plurality of erase units, each of them with a size of 64 kbytes], **each erase unit including a plurality of programming units arranged in a sequence** [The corresponding programming units are the sectors: "Though read and write can be performed in the unit of a sector (512 bytes), erase can only be performed in the unit of a relatively large size such as 64 kbytes (column 1, lines 33-35)"; since each erase unit is 64 kbytes in size and each sector has 512 bytes, an erase unit includes a plurality of 128 sectors (i.e., the programming units); figure 3 shows that sectors 0, 1, 2 and 3 of

PGN=X are arranged in a sequence (column 3, lines 26-37)], **each programming unit comprising a plurality of non-volatile memory cells** [a sector has 512 bytes (column 1, lines 33-35), note that each "bit" of the 512 bytes represents a memory cell] **wherein a logical unit is associated with a first erase unit and includes a plurality of logic sub-units arranged in a sequence** [figures 4 and 6 show the correspondence between logical entities and physical entities; column 3, lines 30-37; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7), note that each "logical sector number" represents a "logic sub-unit," and the collection of all logical sector numbers associated with a block represents a "logical unit"], **the method comprising:**

grouping the programming units [The corresponding programming units are the sectors (Though read and write can be performed in the unit of a sector (512 bytes) (column 1, lines 33-35))] **within the first erase unit into a plurality of groups of programming units** [A semiconductor storage device comprises a flash memory and manages the sectors in the unit of groups each consisting of n sectors (abstract); The semiconductor storage device 11 manages four sectors as one group (column 3, lines 26-27); figure 5 shows a plurality of groups in sequence], **the programming units within each group arranged in a sequence comprising a first programming unit, a second programming unit, and a third programming unit** [figure 5 shows a plurality of groups, including two groups PGN=X and PGN=Y; each group of PGN=X and PGN=Y comprises at least 4 programming units, including a first programming unit

(denoted as "0"), a second programming unit (denoted as "1"), and a third programming unit (denoted as "2").

writing contents of a first logical sub-unit into the first programming unit in a first group of programming units of the first erase unit [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 4 into the flash memory (see figures 7A~7B, note that 4 divided by 4 gives a quotient of 1 and a remainder of 0), and if sector 0 of group PGN=1 has not been written with data yet, then the contents of logical sector number 4 (i.e., the first logical sub-unit) from the host will be written into sector 0 (i.e., the first programming unit) of group PGN=1 (i.e., the first group) because this is a "new write" (column 4, line 58 to column 5, line 7; column 5, line 63 to column 6, line 27); figures 5 and 8A~8E]; **and**

then writing contents of a third logical sub-unit into the third programming unit in a first group of programming units [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 6 into the flash

memory (see figures 7A~7B, note that 6 divided by 4 gives a quotient of 1 and a remainder of 2), and if sector 2 of group PGN=1 has not been written with data yet, then the contents of logical sector number 6 (i.e., the second logical sub-unit) from the host will be written into sector 2 (i.e., the second programming unit) of group PGN=1 (i.e., the first group) because this is a "new write" (column 4, line 58 to column 5, line 7; column 5, line 63 to column 6, line 27); figures 5 and 8A~8E], **the third logical sub-unit later in the sequence of logical sub-units than the first logical sub-unit, and the third programming unit later in the sequence of the programming units within the first group than the first programming unit and the second programming unit** [logical sector number 6 (i.e., the second logical sub-unit) is later than logical sector number 4 (i.e., the first logical sub-unit); figure 5 shows a plurality of groups, including two groups PGN=X and PGN=Y; each group of PGN=X and PGN=Y comprises at least 4 programming units, including a first programming unit (denoted as "0"), a second programming unit (denoted as "1"), and a third programming unit (denoted as "2")],

then writing contents of a second logical sub-unit of the first logic unit into a programming unit in a second group of programming units of the first erase unit [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of

logical sector number 5 into the flash memory (see figures 7A~7B, note that 5 divided by 4 gives a quotient of 1 and a remainder of 1), and if sector 1 of group PGN=1 has already been written with data, then the contents of logical sector number 5 (i.e., the third logical sub-unit) from the host will be written into sector 1 of group PGN=2 (i.e., the second group) because this is a "additional write" (column 4, line 58 to column 5, line 28; column 6, lines 29-61); figures 5 and 8A~8E], **the second logical sub-unit earlier in the sequence of logical sub-units than the third logical sub-unit** [logical sector number 5 (the third logical sub-unit) is earlier than logical sector number 6 (i.e., the second logical sub-unit)].

Regarding claim 1, Miyauchi does not teach that **the non-volatile memory is an MLC NAND memory**.

However, the invention of Miyauchi is directly applicable to any type of flash memories, including NAND flash memory and MLC NAND flash memory.

Further, Kramer teaches in the invention "Fault-Tolerant Codes for Multi-Level Memories" a fault-tolerant code semiconductor flash memory storage devices including an array of individual multi-level cell (MLC) storage devices [abstract; column 2, lines 1-15] as well as NAND flash memory [column 2, lines 36-57].

Kramer also teaches that the motivation of using a MLC NAND flash non-volatile memory is to take advantage the benefits that it offers [A flash memory with multilevel cell significantly reduces the memory per-bit cost. For instance, a 32 Mb multilevel cell (MLC) Flash memory storing two bits of data per cell achieves 32 Mb memory storage capacity using 16M Flash memory cells (column 2, lines 5-10); a 128 Mb multi-level

NAND flash memory stores two bits per cell by tight programmed cell threshold voltage (V_{th}) control. Program states have 0.4V V_{th} distribution and a 0.8V separation gap. Incremental-step pulse programming (ISPP) gives narrower V_{th} distribution with a smaller stepping voltage (column 2, lines 39-44)].

Therefore, it would have been obvious to use a MLC NAND flash non-volatile memory as the type of flash memory of Miyauchi's invention because it reduces the memory per-bit cost and gives narrower V_{th} distribution with a smaller stepping voltage.

As to claim 3, Miyauchi teaches that **the method of claim 1, further comprising:**

After the step of writing contents of a second logic sub-unit, writing contents of a fourth logical sub-unit into a programming unit of the second group, the fourth logical sub-unit later in the sequence of logical sub-units than the second logical sub-unit [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 7 into the flash memory (see figures 7A~7B, note that 7 divided by 4 gives a quotient of 1 and a remainder of 3), and if sector 3 of group PGN=1 has already been written with data, then logical sector number 7 (i.e., the fourth logical sub-unit) from the host will be written into sector 3 of group PGN=2 (i.e.,

the second group) because this is a "additional write" (column 4, line 58 to column 5, line 28; column 6, lines 29-61); logical unit number 7 (i.e., the fourth logical sub-unit) is later than logical unit number 5 (i.e., the third logical sub-unit); figures 5 and 8A~8E].

As to claim 4, Miyauchi teaches that **the method of claim 1, further comprising:**

writing a first identifier into the first group of programming units [figure 5 shows that the first group has an identifier PGN (Physical Group Number)=X and the second group has an identifier PGN=Y; figures 8A~8E and 9A~9E show more examples; the Logical Group Number is stored in the group management information (figure 2, 22; column 4, lines 1-25)], **the first identifier being arranged to substantially identify the first logical sub-unit** [figures 5 and 6 show the correspondence between the identifiers and the logical sub-units; a logical sector number is converted to a logical group number and an offset address, and a conversion table is provided to convert a logical group number to a physical group number (abstract)];

writing contents of a first logical sub-unit of a second logical unit into a programming unit of a second group of programming units in the first erase unit [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 8 into the flash memory (see figures 7A~7B, note that 8 divided

by 4 gives a quotient of 2 and a remainder of 0), and if sector 0 of group PGN=2 has not been written with data yet, then the contents of logical sector number 8 (i.e., the first logical sub-unit) from the host will be written into sector 0 (i.e., the first programming unit) of group PGN=2 (i.e., the second group) because this is a “new write” (column 4, line 58 to column 5, line 7; column 5, line 63 to column 6, line 27); figures 5 and 8A~8E]; and

writing a second identifier into the second group of programming units [figure 5 shows that the first group has an identifier PGN (Physical Group Number)=X and the second group has an identifier PGN=Y; figures 8A~8E and 9A~9E show more examples; the Logical Group Number is stored in the group management information (figure 2, 22; column 4, lines 1-25)], **the second identifier being arranged to substantially identify the second logical sub-unit** [figures 5 and 6 show the correspondence between the identifiers and the logical sub-units; a logical sector number is converted to a logical group number and an offset address, and a conversion table is provided to convert a logical group number to a physical group number (abstract)].

As to claim 5, Miyauchi teaches that **each of the erase units is a physical block, and the first logic unit is a logic block** [figures 5 and 6 show the correspondence between the physical entities and the logical sub-units; column 3, line 66 to column 4, line 40], **wherein each of the programming units is a physical page** [the sectors (column 3, lines 42-52)]; **and wherein each of the logical sub-units is a logical page** [the logical sector number from the host (abstract; column 6, lines 4-7)].

As to claim 10, the limitations recited in this claim are substantially the same as those recited in claims 1 and 4. Thus claim 10 is rejected by the same reasons as applied to claim 1 and claim 4. Refer to "As to claim 1" and "As to claim 4" presented earlier in this section for details.

As to claim 15, Miyauchi teaches that **the non-volatile memory system** [figure 18 shows the memory system], **wherein the controlling means comprises: code devices for performing the sequence of operations** [figures 7A~7B and figure 10 show the code that control the operation of the storage device; the code device comprises memory that stores the code and the address conversion table (figure 1, 16)]; **and a processor for accessing and executing the code devices** [CPU (figure 1, 13) is the processor; the CPU controls the operation of the entire semiconductor storage device (column 3, lines 24-25)].

As to claim 19, Miyauchi teaches that **each of the erase units is a physical block, and the first logic unit is a logic block** [figures 5 and 6 show the correspondence between the physical entities and the logical sub-units; column 3, line 66 to column 4, line 40], **wherein each of the programming units is a physical page** [the sectors (column 3, lines 42-52)]; **and wherein each of the logical sub-units is a logical page** [the logical sector number from the host (abstract; column 6, lines 4-7)].

As to claim 22, it recites substantially the same limitations as in claim 1, and is rejected for the same reasons set forth in the analysis of claim 1. Refer to "As to claim 1" presented earlier in this Office Action for details.

As to claim 43, Miyauchi teaches that **the method of claim 1, further comprising:**

after the step of writing contents of the third logical sub-unit, receiving contents of a fourth logical sub-unit to be written into the non-volatile memory, the fourth logical sub-unit earlier in the sequence of logical sub-units than the third logical sub-unit [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the contents of logical sector number 7 into the flash memory (see figures 7A~7B, note that 7 divided by 4 gives a quotient of 1 and a remainder of 3), and if sector 3 of group PGN=1 has already been written with data, then logical sector number 7 (i.e., the fourth logical sub-unit) from the host will be written into sector 3 of group PGN=2 (i.e., the second group) because this is a "additional write" (column 4, line 58 to column 5, line 28; column 6, lines 29-61); logical unit number 7 (i.e., the fourth logical sub-unit) is later than logical unit number 5 (i.e., the third logical sub-unit); figures 5 and 8A~8E];

determining whether one of the groups of programming units in the first erase unit is available for writing into [figure 7A, step S103; column 5, line 63 to column 6, line 28];

responsive to none of the groups of programming units being available for writing into [if the answer is "YES" for step S117 of figure 7B], **merging the contents**

of the first logical unit that were previously written into the plurality of programming units of the first erase unit into programming units of a first group of programming units in a second erase unit [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written data; column 7, lines 20-36]; and then writing the contents of the fourth logical sub-unit into a programming unit of a second group of programming units of the second erase unit [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written (i.e., existing) data; column 7, lines 20-36].

As to claim 44, Miyauchi teaches that **the method of claim 43, wherein the merging step merges the contents of the first logical unit that were previously written into the plurality of programming units of the first erase unit with contents of the first logical unit stored in another erase unit [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written (i.e., existing) data; column 7, lines 20-36].**

As to claim 45, Miyauchi teaches that **the method of claim 43, further comprising: determining whether additional contents of the first logical unit remain to be updated [figures 7A~7B; column 4, line 58 to column 5, line 28];**

responsive to additional contents remaining to be updated, repeating the determining, merging, and writing steps [figure 7A, step S103; column 5, line 63 to column 6, line 28; figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written data; column 7, lines 20-36]; **and**

responsive to additional contents not remaining to be updated, merging the contents associated with the first logical unit and stored in the second erase unit into a third erase unit [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written data; column 7, lines 20-36].

As to claim 46, Miyauchi teaches that **the system of claim 23, further comprising:**

after the step of writing contents of the third logical sub-unit, receiving contents of a fourth logical sub-unit to be written into the non-volatile memory, the fourth logical sub-unit earlier in the sequence of logical sub-units than the third logical sub-unit [figures 7A and 7B show the flowchart of write operation on how the content of a logic sector unit is written into the corresponding physical sector of a particular group; For example, when the logical sector numbers designated by the host 10 are 4, 5 and 6, the logical group number is set as "1" and the offset addresses OA are set and "0", "1" and "2" (column 6, lines 4-7); for example, when the host tries to write the

contents of logical sector number 7 into the flash memory (see figures 7A~7B, note that 7 divided by 4 gives a quotient of 1 and a remainder of 3), and if sector 3 of group PGN=1 has already been written with data, then logical sector number 7 (i.e., the fourth logical sub-unit) from the host will be written into sector 3 of group PGN=2 (i.e., the second group) because this is a "additional write" (column 4, line 58 to column 5, line 28; column 6, lines 29-61); logical unit number 7 (i.e., the fourth logical sub-unit) is later than logical unit number 5 (i.e., the third logical sub-unit); figures 5 and 8A~8E]; **determining whether one of the groups of programming units in the first erase unit is available for writing into** [figure 7A, step S103; column 5, line 63 to column 6, line 28];

responsive to none of the groups of programming units being available for writing into [if the answer is "YES" for step S117 of figure 7B], **merging the contents of the first logical unit that were previously written into the plurality of programming units of the first erase unit into programming units of a first group of programming units in a second erase unit** [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written data; column 7, lines 20-36]; **and**

then writing the contents of the fourth logical sub-unit into a programming unit of a second group of programming units of the second erase unit [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in

figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written (i.e., existing) data; column 7, lines 20-36].

As to claim 47, Miyauchi teaches that **the system of claim 46, wherein the merging step merges the contents of the first logical unit that were previously written into the plurality of programming units of the first erase unit with contents of the first logical unit stored in another erase unit** [figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written (i.e., existing) data; column 7, lines 20-36].

As to claim 48, Miyauchi teaches that **the system of claim 46, further comprising:**
determining whether additional contents of the first logical unit remain to be updated [figures 7A~7B; column 4, line 58 to column 5, line 28];
responsive to additional contents remaining to be updated, repeating the determining, merging, and writing steps [figure 7A, step S103; column 5, line 63 to column 6, line 28; figure 7B, steps S120 to S126; column 6, lines 29-61; the merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written data; column 7, lines 20-36]; **and**
responsive to additional contents not remaining to be updated, merging the contents associated with the first logical unit and stored in the second erase unit into a third erase unit [figure 7B, steps S120 to S126; column 6, lines 29-61; the

merge operation is further illustrated in figures 8A~8E, where contents of a logical unit is merged into other groups/erase-units with previously written data; column 7, lines 20-36].

As to claim 49, Miyauchi teaches that **the method of claim 1, wherein the step of writing contents of the second logical sub-unit writes the contents into a second programming unit in the second group of programming units of the first erase unit, the second programming unit later in the sequence of programming units than a first programming unit in the second group, and earlier in the sequence of programming units than a third programming unit in the second group** [Figure 5 of Miyauch explicitly show a plurality of groups (i.e., blocks) of programming units. For example, group PGN=X comprises four programming unites denoted as 0, 1, 2, and 3. Similarly, group PGN=Y comprises four programming unites denoted as 0, 1, 2, and 3. The write operations is further illustrated in figures 7A and 7B to explain when a logical unit of a “out-of-sequence writes” operations is destined in either group PGN=X or group PGN=Y (for example, figure 7A, step S104, “search a group not used;” figure 7B, step S120, “search a group not used”)].

As to claim 50, Miyauchi teaches that **the system of claim 22, wherein the step of writing contents of the second logical sub-unit writes the contents into a second programming unit in the second group of programming units of the first erase unit, the second programming unit later in the sequence of programming units than a first programming unit in the second group, and earlier in the sequence of programming units than a third programming unit in the second**

group [Figure 5 of Miyauch explicitly show a plurality of groups (i.e., blocks) of programming units. For example, group PGN=X comprises four programming unites denoted as 0, 1, 2, and 3. Similarly, group PGN=Y comprises four programming unites denoted as 0, 1, 2, and 3. The write operations is further illustrated in figures 7A and 7B to explain when a logical unit of a "out-of-sequence writes" operations is destined in either group PGN=X or group PGN=Y (for example, figure 7A, step S104, "search a group not used;" figure 7B, step S120, "search a group not used")].

8. *Related Prior Art On Record*

The following list of prior art is considered to be pertinent to applicant's invention, but not relied upon for claim analysis conducted above.

- Marshall et al., (US 5,832,493), "Flash File management System."
- Tomori et al., (US patent Application Publication 2002/0085433), "Data management System and Data management Method."
- Achiwa et al., (US 5,737,742), "memory system Using Flash memory and Method of Controlling the memory System."
- Sukegawa, (US 5,572,466), "Flash Memory Chips."
- Hong et al., (US 6,236,593), "Method for Simultaneously Programming Plural Flash Memories Having Invalid Blocks."
- Hewitt et al., (US 5,488,711), "Serial EEPROM device and Associated Method for Reducing Data Load Time Using a Page Mode Write Cache."
- Lofgren et al., (US 6,230,233), "Wear Leveling Techniques for Flash EEPROM Systems."

- Jou et al., (US 5,568,423), "Flash Memory Wear Leveling System Providing Immediate Direct Access to Microprocessor."
- Miyauchi, (US 6,970,890), "Method and Apparatus for Data Recovery."
- Sukegawa et al., (US 5,603,001), "semiconductor Disk System Having a Plurality of Flash Memories."
- Norman, (US 5,754,567), "Write Reduction in Flash Memory systems through ECC Usage."
- Bruce et al., (US 6,000,006), "Unified Re-Map and Cache-Index Table with Dual Write-Counters for wear-Leveling of Non-Volatile Flash RAM Mass Storage."

Conclusion

9. Claims 1, 3-5, 10, 15, 19, 22 and 43-50 are rejected as explained above.

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sheng-Jen Tsai whose telephone number is 571-272-4244. The examiner can normally be reached on 8:30 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Sheng-Jen Tsai
Examiner
Art Unit 2186

January 9, 2008



MATTHEW KIM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100